

Virtual Path Assignment in ATM Networks

Metin Aydemir and Yannis Viniotis

Department of Electrical and Computer Engineering

North Carolina State University, Raleigh, NC 27695-7911, USA.

Phone: (919) 515-5148. Fax: (919) 515-2285.

email: {metin,candice}@ecec.v.ncsu.edu

Abstract

Virtual Path (VP) Assignment is an important resource management activity in ATM networks. By using the VP concept, the logical layout of the network can be administered to minimize the total network cost to the provider. We formulate the VP Assignment problem as a constrained optimization problem; the objective function to be minimized reflects processing/control costs at nodes and transmission costs between nodes. The quality of service requirements and overall throughput of the network are taken into consideration in the solution process as constraints.

Since the solution space is complex, we develop a heuristic algorithm based on descent methods to reach a “near-optimal” VP configuration for a given physical network and traffic demand. The Multi-Rate Loss Network model is used in the calculation of connection-level blocking probabilities. Results for static connection-routing are presented, for various forms of the objective function and constraints.

Keywords

ATM, virtual path, optimization, loss networks, quality of service, high-speed networks.

1 INTRODUCTION

The Asynchronous Transfer Mode (ATM) is accepted as the transport technology for Broadband Integrated Services Digital Networks (B-ISDN), due to its flexibility in accommodating a wide range of services with different traffic characteristics, and due to its efficient use of transport capacity by means of statistical multiplexing, among other advantages (ITU-T Recommendation I.150, 1991). The Virtual Path concept in ATM Networks is developed in order to achieve lower total network costs by reducing the processing and control costs (Kanada *et al.*, 1987) (Tokizawa *et al.*, 1988) (Addie *et al.*, 1988).

The Virtual Path concept is accommodated into the ATM Specifications by the ITU-T standards body (ITU-T Recommendation I.311, 1991). The Virtual Paths are logical paths defined at the transmission path level (Physical Layer) of ATM. In ITU-T terminology, a Virtual Path is defined as a labeled path (bundled multiplexed virtual channels or

connections) between its end-nodes (also called virtual path terminators). The virtual path concept and implementation issues are addressed in detail in Sato *et al.* (1990), and Burgin and Dorman (1991).

The definition of VPs is refined by associating a deterministic capacity with the virtual path. This reservation of capacity for VPs has two important consequences: i) the statistical multiplexing is limited to the traffic within a virtual path; ii) a logical network (VP Network) emerges on top of the physical network topology.

Assuming that all connections are made by using VPs, this logical VP Network is a very flexible tool that can be employed in the management of network resources. The configuration of the VP Network has a major impact on the processing/control costs and transmission costs of a network.

VPs impact the network processing and control costs in three areas: call acceptance and call set-up, cell routing, and adaptability to changes. Employing VPs reduces the processing and delay associated with the call acceptance control (CAC) functions. Since VPs have guaranteed bandwidth along their path, these functions have to be performed only at the beginning node of a VP. When a call is active, only the VPI field of each ATM cell is processed in intermediate nodes of the VP. The cell processing time is reduced compared to VC Switching, which uses both VPI and VCI fields. Varying traffic conditions and network failures can be managed by adjusting the VP Network to accommodate the changing conditions and to maintain network performance.

VPs increase network transmission costs and decrease network throughput. The VP concept is based on reserving capacity in anticipation of expected traffic, and this results in decreased utilization of available transmission capacity. Allocating the capacity of a physical link to VP's using it reduces the sharing of the transmission capacity resource. The application of VP concept reduces the link utilization and overall throughput of the network, compared to total sharing (no VP) case.

We can consider two possible VP Assignment cases to illustrate the effect of the VP Network on the network performance. First, consider a fully-connected VP Network. A VP Network can be formed such that every node has one or more logical link (VP) to every other node. In this network, connections are established using only one VP, thus minimizing control costs. Applicability of this scenario decreases as the number of nodes are increased. For a sufficiently large network, the VPI field in ATM Cell header renders this solution impossible, due to the fact that the size of this field imposes a limit on the number of VPs associated with one node. The proliferation of VPs has a negative effect on the network utilization. For most network topologies, the reduced throughput will probably be prohibitive, even before any other factor comes into play. Second, consider the case where the VP Network is the same as the physical ATM Network. In this case, the VP Network is such that each physical link in the ATM Network contains only one VP, which carries all connections in the link. This is equivalent to not using the VP feature at all. In this case, the network utilization is maximized, and traffic will observe the lowest connection blocking probability. However, the connection establishment activity and processing at the intermediate nodes are at their highest cost. In general, an optimum VP Network solution lies between the two extreme VP Assignment cases considered above. Its exact location is a function of how we define optimality, the physical network under study, and the traffic flows in that network. This paper contributes to the existing work in this subject by proposing a new heuristic search method that is intuitively simple, for the solution of VP assignment problem.

The rest of the paper is organized as follows. In Section 2, we define the details of the problem definition and methods chosen on attacking the problem. Section 3 addresses the computationally challenging blocking probability calculation in a VP Network. In Section 4, we describe our proposed search method, used in the VP Assignment Algorithm. Initial results from test runs of the algorithm and some conclusions are presented in Sections 5 and 6.

2 PROBLEM FORMULATION

We consider the problem of finding a VP Network (i.e. assigning VPs and allocating their capacities) which minimizes an objective function subject to a set of QoS related constraints, for a given network topology, link capacities, and traffic demands.

The traffic demand is defined in terms of *traffic streams*. A *traffic stream* is a distinct traffic flow between two nodes, with defined call arrival rate, call holding time, and cell arrival rate. The cell arrival rate can be the average cell arrival rate if a constant bit rate (CBR) source is considered, or it can be an equivalent rate for bursty sources based on their quality of service requirements. Users can be categorized into different classes with distinct holding time statistics and bandwidth requirements. This way, different traffic types can be modeled with different *traffic streams*. This multi-class traffic representation provides us with a method to define the traffic demand in detail.

The solution of the VP Assignment problem is highly dependent on the solution of the connection routing problem and vice versa. The following argument gives insight into this interdependency. One needs to know, at least statistically, the traffic offered to various paths in order to accurately judge the merit of allocating a VP on the path. This requires selection of physical routes for traffic streams before the VP Assignment Algorithm proceeds. On the other hand, the connection routing problem can be solved most effectively when the final logical VP Network is known and available.

Thus, the problem can be approached in three ways:

1. Assign VP end nodes, and determine traffic streams that belong to each VP, then select routes for VPs (Chlamtac *et al.*, 1993). This method precludes the possibility of using multiple VP's on a connection.
2. Assign the VP's and connection routes simultaneously within the same procedure (Lin and Cheng, 1993). This formulation yields a complex problem, which cannot be solved directly (Cheng and Lin, 1994).
3. Select physical routes for each traffic stream, then form the VP Network based on this traffic layout.

In this paper, we assume that physical routes for traffic streams are provided (last approach in the above list). This approach is chosen as a first attempt to tackle the problem, due to its simplicity.

We make certain assumptions regarding the network structure, which we summarize below. No restrictions are imposed on the topology of the network. VPs can be established between any two nodes of the network. The nodes of the network are functionally identical. All nodes can be the source or destination of the network traffic. Nodes are not classified as ATM Cross-Connect or ATM Switching nodes. The assumptions on the

node characteristics are adopted in order to provide an unrestricted solution space for the VP assignment problem. The physical links of the network are bidirectional; the capacity available on a link can be used in either direction. These assumptions give us a realistic and flexible model, which can be applied to a wide range of practical networks.

We also make the following assumptions on the connection establishment/handling: i) each connection can use one or more VPs. Allowing more than one VP on a single physical route of a connection results in a larger solution space, ii) the same route is used by connections of a traffic stream. A fixed routing scheme is assumed for simplicity. This assumption does not preclude other possible routing schemes at the call acceptance time, iii) each connection request has an associated equivalent bandwidth based on the traffic stream it belongs to. This bandwidth is allocated to the connection deterministically, when the connection is accepted, iv) if a connection request cannot be satisfied, due to capacity constraints, then the connection request is rejected. This event is called connection blocking, and the probability of call blocking is the same for all connections of a certain traffic stream.

The attributes of Virtual Path j are represented by the triplet $\mathbf{V}_j = \{V_j^{pa}, V_j^{ts}, V_j^c\}$. In this triplet, V_j^{pa} represents the set of physical links in the path of the virtual path j . V_j^{ts} denotes the set of traffic streams using the virtual path j . V_j^c denotes the capacity allocated to the virtual path j . The set of all virtual paths is denoted by $\mathbf{V} = \{\mathbf{V}_j; j = 1, 2, \dots, J\}$. The size J of \mathbf{V} is an output variable, since the number of virtual paths in the system will depend on the solution of VP Assignment problem.

The following two VP characteristics are also assumed : 1) VP's are unidirectional logical links. Since the traffic streams are also defined as one-way traffic, the flow of traffic streams of a VP should match the direction of that VP. 2) Virtual Paths reserve the capacity allocated to them deterministically (i.e., statistical multiplexing of VP's is not considered). Statistical multiplexing within a VP is of course allowed. This is implicit in the equivalent bandwidth concept used in the characterization of traffic streams.

We assume that the VP Assignment Algorithm will be used off-line, at a central point for the whole network. We also assume that there will not be real-time constraints, thus the algorithm can be computationally intensive.

Based on the assumptions and choices made above, we can now define the input and output variables of the VP Assignment Algorithm. Traffic streams are input variables, defined by their connection arrival rate λ_r , bandwidth requirement b_r , and mean connection holding time $1/\mu_r$, where $r = \{1, 2, \dots, R\}$. The physical route T_r of stream r is also an input parameter, since we assume that traffic routes are already selected. T_r is a set of links that connect the source to the destination node. The network topology is specified by the node pairs that are connected by physical link i , and the capacity C_i of physical link i , where $i = \{1, 2, \dots, I\}$. The output of the VP Assignment Algorithm is the set of all VPs (\mathbf{V}).

In the next subsection, we concentrate on the selection of the objective function and constraints for the optimization problem.

2.1 Objective Function and Constraints

The objective function should reflect all parameters that are taken into account when considering the network cost. In this stage, we opted to use very simple objective functions due to the need to calculate their value repetitively. First, we choose to minimize the

number of logical links used by a connection as a measure of control and processing costs. Secondly, since the network topology and transmission capacity is fixed and given, we try to balance maximizing the utilization of the network versus minimizing the control and processing costs.

Before considering variations of objective functions further, let us define the constraint parameters of this problem. The physical limitations of the network are obvious constraints in the optimization procedure. The sum of capacities allocated to VP's that are using physical link i must be less than or equal to the physical link capacity C_i . For link i , this constraint can be formulated as:

$$\sum_{\forall j:i \in V_j^{pa}} V_j^c \leq C_i \quad (1)$$

where V_j^{pa} is the set of physical links used in virtual path j , and V_j^c is the capacity assigned to virtual path j . Inequality (1) is enforcing the capacity limitations of the network. It is obvious from (1) that we consider policies where the allocated VP capacities are guaranteed to be available (i.e. statistical multiplexing at the VP level is not performed).

There is a set of VPs for each traffic stream r , such that this set covers the physical path T_r , of stream r . For stream r , this constraint can be written as:

$$\bigcup_{\forall j:r \in V_j^{ts}} V_j^{pa} = T_r, \quad (2)$$

where V_j^{ts} is the set of traffic streams using the virtual path j . Equation (2) provides that there is a path in the VP Network to carry each defined traffic stream, and no traffic stream is left out in the VP Network.

The quality of service requirements of various classes can be implemented either as constraints, or as part of the objective function. We follow both approaches here. Of course, a variety of QoS measures can be defined in a network. For simplicity, we deal only with the following QoS measures: 1) *Blocking of Calls*: The blocking probability of traffic stream r , call it $\beta_r(\mathbf{V})$, must be less than or equal to ϵ_r , the maximum blocking probability allowed. In general, $\beta_r(\mathbf{V})$ for a given stream r will only depend on the subset of virtual paths used by traffic stream r , and the other traffic carried on these virtual paths. 2) *Number of Hops*: H_r , the number of VPs traversed by a connection of traffic stream r between its source and destination must be less than K , where K is a given positive number. By imposing this limitation, we bound the connection set-up time for all traffic streams. Note that this constraint does not put a bound on the maximum delay a connection will encounter, since a VP may traverse more than one physical link.

Based on the above, the following three objective functions are considered in this paper:

1. Minimum hops with connection blocking as a constraint

$$\min_{\{\mathbf{V}\}} \sum_{\tau=1}^R H_{\tau}, \quad (3)$$

subject to:

$$\sum_{V_j: i \in V_j^{pa}} V_j^c \leq C_i, \quad \forall i \in \{1, 2, \dots, I\}, \tag{4}$$

$$\bigcup_{V_j: r \in V_j^{ts}} V_j^{pa} = T_r, \quad \forall r \in \{1, 2, \dots, R\}, \tag{5}$$

$$\beta_r(\mathbf{V}) \leq \varepsilon_r, \quad \forall r \in \{1, 2, \dots, R\}, \tag{6}$$

$$H_r(\mathbf{V}) \leq K, \quad \forall r \in \{1, 2, \dots, R\}, \tag{7}$$

where H_r is the number of VPs used by the traffic stream r .

In the objective function of equation (3), the total number of hops experienced by all traffic streams is minimized. The integer-valued sum of equation (3) is an indicator of processing costs over the entire network, which is minimized while blocking probabilities are kept at acceptable level by means of inequality (6).

2. Minimum hops (weighted) with connection blocking as a constraint

$$\min_{\{\mathbf{V}\}} \sum_{r=1}^R \lambda_r \cdot H_r, \tag{8}$$

subject to:

$$\sum_{V_j: i \in V_j^{pa}} V_j^c \leq C_i, \quad \forall i \in \{1, 2, \dots, I\}, \tag{9}$$

$$\bigcup_{V_j: r \in V_j^{ts}} V_j^{pa} = T_r, \quad \forall r \in \{1, 2, \dots, R\}, \tag{10}$$

$$\beta_r(\mathbf{V}) \leq \varepsilon_r, \quad \forall r \in \{1, 2, \dots, R\}, \tag{11}$$

$$H_r(\mathbf{V}) \leq K, \quad \forall r \in \{1, 2, \dots, R\}, \tag{12}$$

where λ_r is the connection arrival rate of traffic stream r .

Equation (8) is a measure of number of hops experienced by an arbitrary connection. Since the intensity of traffic is taken into account in equation (8), it is a more precise minimization compared to equation (3). Equation (8) is a real-valued sum, thus computationally more expensive. Since the parameters $\{\lambda_r\}$ are fixed for a given network, objective function (8) is not a special case of objective function (13). The VP networks obtained from the two formulations will in general be different.

3. Balancing control and transmission costs

$$\min_{\{\mathbf{V}\}} \left\{ \alpha \sum_{j=1}^J D_j + \sum_{r=1}^R H_r \right\}, \tag{13}$$

subject to:

$$\sum_{V_j: i \in V_j^{pa}} V_j^c \leq C_i, \quad \forall i \in \{1, 2, \dots, I\}, \tag{14}$$

$$\bigcup_{j:r \in V_j^{\text{pa}}} V_j^{\text{pa}} = T_r, \quad \forall r \in \{1, 2, \dots, R\}, \quad (15)$$

$$\beta_r(\mathbf{V}) \leq \varepsilon_r, \quad \forall r \in \{1, 2, \dots, R\}, \quad (16)$$

$$H_r(\mathbf{V}) \leq K, \quad \forall r \in \{1, 2, \dots, R\}, \quad (17)$$

where D_j is the number of VPs in physical link j , and H_r is the number of VPs used by traffic stream r . As we discussed earlier in this section, J is an output variable, the value of which is determined as part of the solution. The value of the input parameter α is chosen based on the network characteristics. Constraint (16) guarantees an acceptable level of blocking, and constraint (17) makes sure the call set-up delays are bounded.

Equation (13) is a simple balancing formula for two reasons: i) Maximize the sharing of the VP's, so that network utilization is maximized and blocking of connections is minimized (first sum); ii) Minimize the number of hops in a connection so that the set-up costs are low (second sum).

Problems (3), (8), and (13) are hard optimization problems, due to their nonlinearity and complexity of the constraint spaces. There are two reasons for this complexity. First, some of the optimization variables are integer-valued. Second, the constraints in inequalities (6), (11), and (16) are not convex, and therefore traditional solution techniques are not applicable.

The calculation of blocking probability $\beta_r(\mathbf{V})$ is difficult in most cases, and an exact method is not available. In the next section, we describe two approximations for this calculation, as a function of VP Network selection.

3 MODEL FOR CONNECTION-LEVEL BLOCKING PROBABILITY CALCULATION

In this section, we describe the product-form model for blocking probability calculation, and two approximate methods that are employed in the VP Assignment algorithm.

For a given VP configuration in an ATM Network, we can view each VP as a logical link in the network. This new logical network topology is referred to as the *VP Network*. Let's assume that a total of R traffic streams exist. The connection requests of stream r are assumed to arrive following a Poisson process with rate λ_r . The connection holding time distribution is assumed to be of phase-type, with mean $(1/\mu_r)$. A connection request that cannot be satisfied (not enough capacity in one of the links on its path), will be cleared without any other consequences. This type of networks has been named *multi-rate loss networks* (Chung and Ross, 1993), and analyzed in the context of circuit-switched telephony networks (Kelly, 1986). Multi-Rate Loss Networks have been shown to fit into a generalized BCMP Network model (Baskett *et al.*, 1975) (Lam, 1977). The equilibrium state probability distribution of this model has a product-form solution.

We next provide the state definition, other parameters, and the set of feasible states as an introduction to the approximation schemes that will follow. The logical link capacities are denoted by $\mathbf{V}^c = (V_1^c, V_2^c, \dots, V_J^c)$, where J is the total number of logical links (VPs). The matrix \mathbf{A} is constructed such that the element $A_{j,r}$ of the matrix denotes the bandwidth or cell arrival rate required by the traffic stream r , on logical link j . The dimension

of \mathbf{A} is $J \times R$. Typically, a traffic stream has a constant bandwidth demand, c , on all of its links, thus a row of matrix \mathbf{A} takes values of zero or c . The logical link capacities and connection bandwidth requirements are integer multiples of a unit capacity.

The state of the system is defined as $\mathbf{n} = (n_1, n_2, \dots, n_R)$, where n_r is the number of active connections of stream r . The set of feasible states $\mathcal{S}(\mathbf{V}^c)$ is given by:

$$\mathcal{S}(\mathbf{V}^c) = \{\mathbf{n} : \mathbf{A} \cdot \mathbf{n} \leq \mathbf{V}^c\}.$$

This representation encapsulates the constraints imposed on the state space due to link capacities. The j th row of the vector inequality $\mathbf{A} \cdot \mathbf{n} \leq \mathbf{V}^c$ provides the constraint imposed by the capacity of link j :

$$A_{j1} \cdot n_1 + A_{j2} \cdot n_2 + \dots + A_{jR} \cdot n_R \leq V_j^c.$$

The state probability distribution is in product-form:

$$P(\mathbf{n}) = \frac{1}{G(\mathbf{V}^c)} \cdot \prod_{i=1}^R \frac{a_i^{n_i}}{n_i!},$$

where $G(\mathbf{V}^c)$ is the normalization constant for a network with capacity vector \mathbf{V}^c and $a_i \stackrel{\text{def}}{=} \lambda_i / \mu_i$. $G(\mathbf{V}^c)$ is given by:

$$G(\mathbf{V}^c) = \sum_{\mathbf{n} \in \mathcal{S}(\mathbf{V}^c)} \left\{ \prod_{i=1}^R \frac{a_i^{n_i}}{n_i!} \right\}.$$

The blocking probability, $\beta_r(\mathbf{V})$ of connections belonging to traffic stream r can be represented by:

$$\beta_r(\mathbf{V}) = 1 - \frac{G(\mathbf{V}^c - \mathbf{A}_r)}{G(\mathbf{V}^c)},$$

where \mathbf{A}_r is the r -th column of matrix \mathbf{A} .

Although an explicit formula is available, the calculation of the normalization constant proves difficult as the network size increases. An efficient method to compute the exact probability distribution of large multi-rate loss networks is not known. In the following subsections, two methods to calculate *approximate* blocking probabilities are presented. These approximations are all variations of the *reduced load approximation* principle, first introduced by Whitt (1985) for single-rate loss networks. They are presented in order of increasing complexity.

3.1 Erlang Fixed Point (EFP) Approximation

If one assumes that blocking between links and between streams of a link, occurs independently, then the blocking probability $\beta_r(\mathbf{V})$, can be approximated by :

$$\beta_r(\mathbf{V}) = 1 - \prod_{j=1}^J (1 - L_j(\mathbf{V}))^{A_{jr}}. \quad (18)$$

The $L_j(\mathbf{V})$'s can be considered as the blocking probability of link j for a unit bandwidth request. With the simplifying independence assumption, the traffic offered to a link will be Poisson. The rate of the Poisson traffic will be thinned by blockings occurred in the other links. This view of the offered load being reduced by blocking was first employed by Whitt for single-rate loss networks (Whitt, 1985). The following equations are used to obtain the $L_j(\mathbf{V})$ values :

$$L_j(\mathbf{V}) = E \left[V_j^c, \frac{1}{1 - L_j(\mathbf{V})} \sum_r A_{jr} a_r \prod_i (1 - L_i(\mathbf{V}))^{A_{ir}} \right], \quad (19)$$

where

$$E[C, v] = \frac{v^C}{C!} \left(\sum_{n=0}^C \frac{v^n}{n!} \right)^{-1}. \quad (20)$$

Equation (20) is the Erlang Loss Formula for a link with capacity C and Poisson traffic rate v . Kelly (1986) has shown that these fixed-point equations have a unique solution in the set $[0, 1]^J$. The Erlang Fixed Point Approximation has been shown to be asymptotically correct in a natural limiting regime (as the link capacity and offered loads are increased simultaneously, while their ratio remains the same).

3.2 Knapsack Approximation

Extensions to the Erlang Fixed Point Approximation were developed by S-P. Chung and K.W. Ross (1993), among others. One approach is to incorporate a one-dimensional iteration (Kaufman, 1981) into fixed point equations.

Consider a single-link system with link capacity V_j^c , and traffic streams $r \in V_j^{ts}$. The bandwidth of stream- r connections is A_{jr} , and the offered load of stream- r connections is a_r . Define

$$K_{jr}[V_j^c; a_l, l \in V_j^{ts}] = 1 - \frac{\sum_{n=0}^{V_j^c - A_{jr}} w(n)}{\sum_{n=0}^{V_j^c} w(n)}, \quad r \in V_j^{ts}$$

where

$$w(0) = 1,$$

$$w(n) = \frac{1}{n} \sum_{r \in V_j^c} A_{jr} a_r w(n - A_{jr}), \quad n = 1, 2, \dots, V_j^c.$$

If we assume that connection blocking is independent from link to link, then we can write the following “thinning” approximation for the offered load (a_{lj}) of stream- l connections on link j . Let T_l be the set of links in the path of stream- l . Then we have that:

$$a_{lj} = a_l \cdot \left(\prod_{i \in T_l - \{j\}} (1 - L_{il}) \right). \quad (21)$$

The original offered load a_l is thus reduced due to the blockings experienced on the other links (i.e., the set $T_l - \{j\}$) of stream- l .

Thus, we have the following set of equations, corresponding to each stream r of each link j :

$$L_{jr}(\mathbf{V}) = K_{jr} \left[V_j^c; a_{lj}, l \in V_j^{ts} \right], \quad \forall j, r : A_{jr} \neq 0. \quad (22)$$

$L_{jr}(\mathbf{V})$ is defined as the approximate probability that the available bandwidth on link j is less than A_{jr} . The same argument applied in EFP approximation can be used to show that there is always a solution to the set of equations in (22). However in this case unlike EFP, the solution does not have to be unique. Finally, the blocking probability of stream r , is approximated by:

$$\beta_r(\mathbf{V}) = 1 - \prod_{j \in T_r} (1 - L_{jr}(\mathbf{V})).$$

This approximation takes into account the multi-class characteristics of the links, thus improving on the EFP approximation which lumps all classes into one flow of unit bandwidth.

3.3 Comparison of Approximations

The fastest approximation procedure is the Erlang Fixed Point (EFP) Approximation. Its iteration involves the computation of Erlang-B formula for each link. The experimentation with EFP Approximation reported in Chung and Ross (1993) shows that EFP tends to underestimate the blocking probabilities in some cases. The Knapsack Approximation computes one dimensional Kaufman Iteration for each traffic stream of each link. This approximation takes into account the multi-class characteristics of the links above the EFP approximation which lumps all classes into one flow of unit bandwidth. The drawback of the Knapsack Approximation is the non-uniqueness of the solution. By choosing the initial values of blocking probabilities realistically, we can insure converging to the correct solution, in most cases. In all executions of the Knapsack Approximation, we have never experienced a non-convergent case.

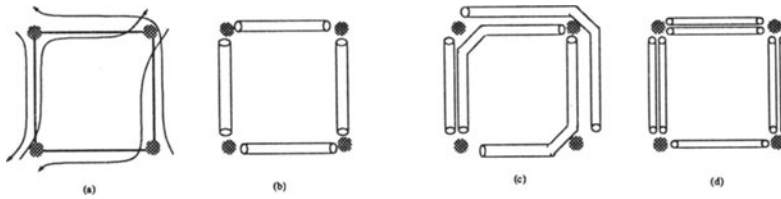


Figure 1 The three extreme solutions: a) Physical network and traffic streams, b) 1-VP/LINK case, c) 1-VP/TS case, d) 1-VP/LINK/TS case.

4 VP ASSIGNMENT ALGORITHM

In this section, we present algorithm VPA, a heuristic VP Assignment algorithm to calculate optimal or near-optimal VP Networks. The current version of the algorithm outputs the following: the final VP allocations and their capacities, the objective function value, the blocking probabilities of traffic streams, and the unused capacities in all physical links.

To achieve this, we define a search method to move around in the domain of valid assignments (VP Networks) and check for optimum solution. Our search method consists of the following three basic steps:

1. Finding a starting point which is a valid solution (a valid VP Network) that satisfies the constraints,
2. Finding incremental changes in the VP Network that achieve a lower value for the objective function, and satisfy the constraints,
3. Repeating the previous step until no more improvement can be found and a termination criterion has been met.

Before describing our method of finding an optimum solution out of all possible VP Networks, let us consider three extreme solutions that give us “boundaries” on the range of possible solutions. These three extreme cases are illustrated in figure 1 for a very simple network, and are described below:

1. Each physical link is a VP. This case is referred to as **1-VP/LINK**. The **1-VP/LINK** corresponds to the maximum sharing case. It yields the lowest blocking probabilities for a fixed connection routing. However, it results in the maximum number of VPs along the path of a connection.
2. Each traffic stream has its own unique VP from source to destination. This case is referred to as **1-VP/TS**. The **1-VP/TS** corresponds to the minimum processing cost, since each connection is using only one VP from source to destination. Conversely, the link sharing is minimized. Each traffic stream has its own reserved capacity along the

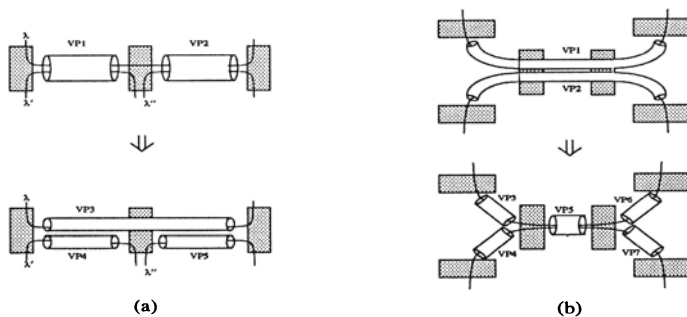


Figure 2 (a) Extending, and (b) Contracting Operations

physical path. The VPs can be shared by the connections of the same traffic stream only.

- Each traffic stream has a unique VP on each physical link of its path from source to destination. This case is referred to as **1-VP/LINK/TS**. The **1-VP/LINK/TS** results in the maximum number of VPs over the whole network. This corresponds to minimum sharing and maximum processing.

An initial solution (a valid VP assignment) has to be chosen as the starting point of the VPA Algorithm. Any VP Network that satisfies the constraints can be the initial solution. We have used the **1-VP/LINK** solution (or a “neighboring” one) as a starting point, depending on the objective function and constraints.

4.1 Finding a Neighboring Assignment

The following two operations (moves) are used to find a neighbor VP Network that is slightly different than the VP Network at hand. Both of these moves involve two VPs at a time.

Extending In the extending operation (see figure 2.a), traffic streams common to two VPs (stream λ in VP1 and VP2) are used to form a new VP (VP3). Non-common traffic streams on the original VPs (λ' and λ'') are carried on separate VPs (VP4 and VP5). In order to apply *extending* to two VPs, the following conditions must be met: i) one VP must start where the other ends, and, ii) two VPs must have at least one common traffic stream.

Contracting In the contracting operation (see figure 2.b), two VPs (VP1 and VP2) are dismantled, and a new merged VP (VP5) is formed on the common physical path of the original VPs. Other VPs (VP3, VP4, VP6, and VP7) are formed for the remaining

physical paths of traffic streams if needed. In order to apply *contracting*, the two VPs must have a common physical link or links that form a path.

Next, we introduce the notion of reachability and show that the contracting and extending operations constitute a sufficient set to traverse the solution space and reach all possible solutions.

Reachability If the topological layout of a valid VP Network, called VPN1, can be obtained by applying predefined operations to another valid VP Network, called VPN2, then VPN1 is *reachable* from VPN2.

Note that only the topological layout of the VP's is considered, while the capacities allocated to VP's are not.

Complete Reachability A set of operations S_{op} , has the *complete reachability property* over a solution set S_{vp} , if any element of S_{vp} is reachable from any other element of S_{vp} by applying a finite number of operations in S_{op} .

In order to show the complete reachability property, we have to show the Reciprocity property, define two special cases of above-mentioned moves, and refine the valid VP Network solution set.

Reciprocity One can easily observe that *extending* and *contracting* are reciprocal operations. From an assignment that has been reached by *extending*, one can go back to the original assignment by one or more *contracting* operations and vice versa.

If an *extending* operation generates only one VP from the original two VPs, it is called a *simple extending* operation. This is possible when the original VPs have the same traffic stream set. Similarly, if a *contracting* operation generates only one VP from the original two VPs, it is called a *simple contracting* operation. This is possible when the original VPs have the exact same path in the physical network.

Restricted VP Network Set The Restricted VP Network set is defined as the set of valid VP Networks where simple contracting and/or simple extending can not be applied to any VP pair.

We will assume that our VP Network solution will be in the Restricted VP Network set. This assumption is justified, because the simple contracting and simple extending operations are desirable in all circumstances. The simple contracting operation improves blocking without affecting the processing costs. The simple extending operation improves processing costs without degrading the blocking probabilities. We can show that if a VP Network solution which may undergo simple extending/contracting operation(s) is chosen, then we will definitely improve the VP Network solution by executing the simple extending/contracting operation(s).

Let us assume now that we apply the simple extending operation to the **1-VP/LINK** solution wherever possible. The resultant VP Network is called "modified **1-VP/LINK**" case. Similarly, the "modified **1-VP/TS**" is the VP Network reached by applying all possible simple contracting operations to **1-VP/TS** case. These modified extreme cases are within our Restricted VP Network set.

Theorem (Complete Reachability). Any element of *Restricted VP Network Set* can be reached by a finite number of successive applications of *extending* and *contracting* operations, starting from an arbitrary *Restricted VP Network Set* element (i.e., the *extending* and *contracting* operations has complete reachability property over the *Restricted VP Network Set*).

Outline of Proof. The following points outline the skeleton of our proof of reachability in Restricted VP Network Set. The complete proof is given in Aydemir and Viniotis (1995).

- Starting with any valid assignment, one can reach one of the modified extreme cases by applying extending or contracting continuously. If extending is applied, the final VP Network will be the “modified 1-VP/TS” case. If contracting is applied, the final VP Network will be the “modified 1-VP/LINK” case.
- Due to the reciprocity property of *extending* and *contracting*, one can reach any valid assignment from one of the modified extreme cases.
- Finally, from a valid assignment one can reach all other valid assignments by successively applying extending and/or contracting. We have shown two possible paths between any two VP Network that pass through the modified extreme cases. □

Once a new VP Network is found, the capacity of new VPs should be assigned and the capacity of existing VPs should be revised. The next section explains our approach to the capacity assignment task.

4.2 Capacity Allocation to VPs

The capacity allocation algorithm is applied to reallocate VP capacities after each move in the VP Assignment Algorithm. The algorithm keeps the blocking probabilities of traffic streams in balance by applying the max-min fairness approach (Bertsekas and Gallager, 1992). This approach can be summarized as trying to improve the blocking probability of the worst-blocked traffic stream by adding capacity to VPs used by this traffic stream.

The algorithm starts with a minimal initial capacity allocation for each VP. The traffic stream with the highest blocking is always chosen and its bandwidth is added to the capacity of all VPs along its path. If physical link capacity limits are reached, no further improvement can be achieved for the VPs involved. The algorithm continues to equally improve the blocking for all traffic streams, until one of the following conditions is met: i) the blocking probability constraints are satisfied for all traffic streams, or ii) no more improvement in blocking is possible, due to link capacity constraints.

5 RESULTS

Preliminary runs with the Virtual Path Assignment (VPA) Algorithm have been executed for various objective functions and constraints for two sample networks (see figure 3). The 10-node network, and the 23-node network topologies are originally presented in Arvidsson (1994) and Siebenhaar (1994), respectively. For each network, we defined traffic streams with fixed routes between every node pair in both directions (90 traffic streams for 10-node, and 506 traffic streams for 23-node network).

The primary outcome of the VPA Algorithm is the VP Network (the set of VPs with assigned capacity and assigned physical path). Additionally, we output the blocking probabilities and number of hops for traffic streams. The unused capacity of physical links at the end of the algorithm is also monitored, even though we do not report it here (see Aydemir and Viniotis, 1995).

We have tested the response of the VPA Algorithm by varying the blocking probability

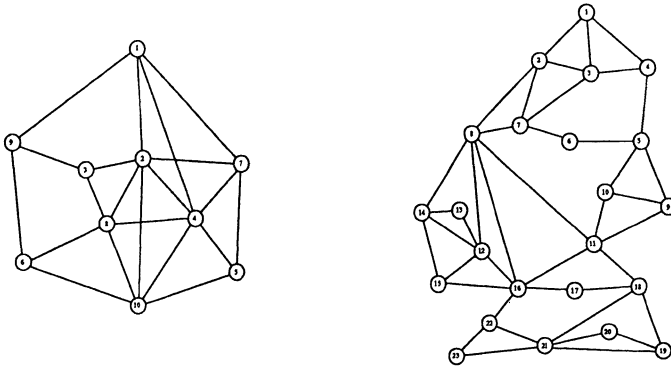


Figure 3 The 10-node and 23-node networks used in the test runs.

constraint, ϵ_r , and maximum hop constraint (K). We have also varied the total offered load on a fixed network topology, and the proportionality constant α in objective function (13). The blocking probability constraint ϵ_r value of 0.01 is applied in all of the experiments except for figure 6.

In figure 4, we demonstrate the accuracy of the EFP and Knapsack Approximations with respect to simulation values. The points in the plot for EFP and Knapsack Approximations are mean error percentages of approximate stream blocking probabilities with respect to a simulation estimate of stream blocking probabilities. The points for the simulation are the average of error percentages of upper bound of confidence interval (95 %) with respect to simulation estimate. The Knapsack Approximation is almost always within the confidence interval of simulation results; the EFP approximation underestimates the blocking probability of connections. In the rest of the runs, the knapsack approximation is employed for blocking probability calculations.

In figure 5, we present the effect of maximum hop constraint (K) on the minimum hop and weighted min hop objectives. Limiting the number of hops experienced, has resulted in achieving better optimization results for the same load conditions. Starting solutions for each K value were different, and this may have caused the algorithm to reach better solutions for lower K values. The variation in the parameter α and its effect on the optimal value is also illustrated in figure 5. As the α value increases, the emphasis on the minimization of number of hops decreases, resulting in the gradual increase of average number of hops.

Figure 6 illustrates the response of the algorithm to changing blocking probability constraints in both 10-node and 23-node networks. The same ϵ_r value is applied to all traffic streams. In the 10-node network, the weighted and non-weighted versions of the objective function that minimizes the number of hops were used and both generated the same results. As the blocking probability constraint is relaxed, the number of hops used

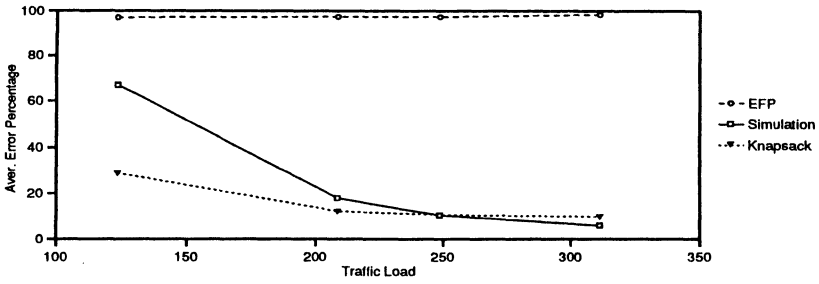


Figure 4 Comparison of Connection Blocking Probability Approximation Methods.

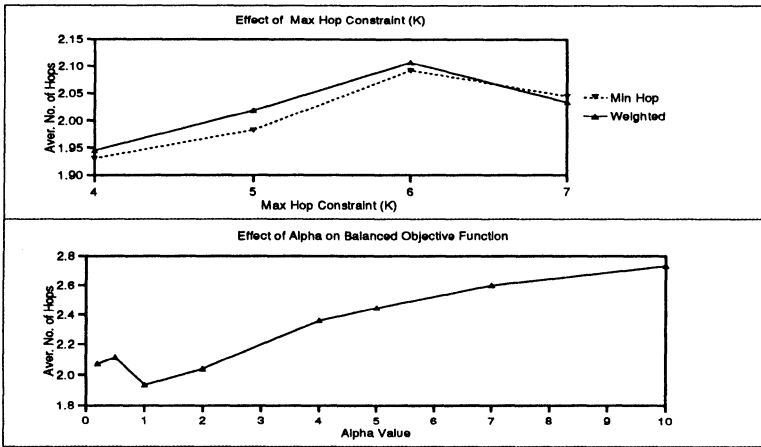


Figure 5 Effects of K (Max Hop Constraint) and α in Balanced Objective.

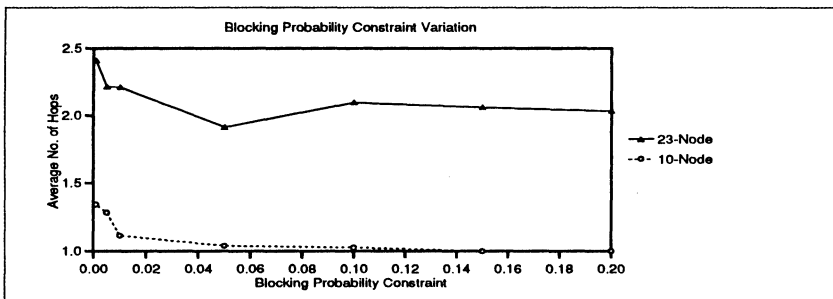


Figure 6 Blocking Probability Constraint (ϵ_r) Variations.

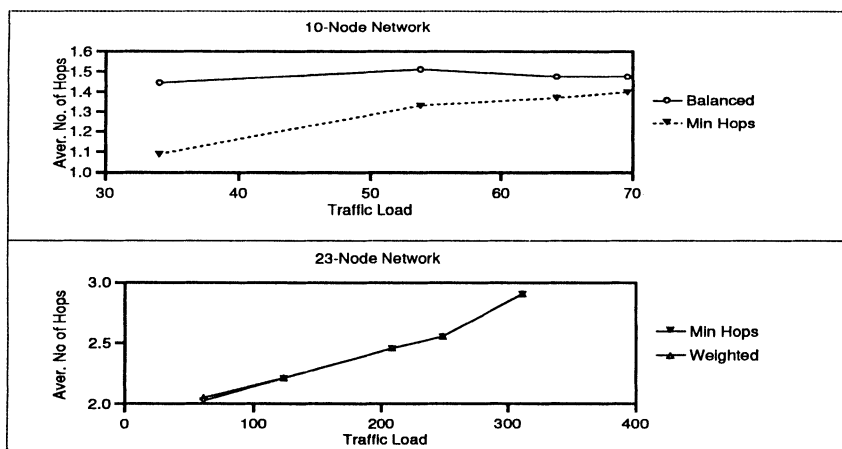


Figure 7 Traffic Load Variations.

by a connection also decreases, until all connections use one VP (1-VP/TS case), in the 10-node network. In the 23-node network, the same trend is observed, except for an unexpected dip at $\epsilon_r = 0.05$.

Figure 7 shows the results of the algorithm for varying traffic loads. The traffic load for the whole network is calculated by using the formula $\sum \lambda_r \cdot b_r$. The mean number of hops experienced by connections and the overall blocking probability of the network are plotted against the network traffic load. This experiment is run with the three objective functions of Section 2.1. The results for weighted and non-weighted minimum hop objective functions were the same. The balanced objective function is run with α set to 0.5. As the traffic load increases on a fixed network, the blocking probability constraint plays a bigger role in all cases. Thus the difference in the outcome diminishes with increasing traffic load. The slope of the increase of mean number of hops is more pronounced in the 23-node network. The minimum hop and weighted minimum hop objectives have performed almost the same, in the large network case too.

6 CONCLUSIONS

In this paper, we presented a heuristic optimization algorithm for the Virtual Path Assignment function in ATM networks. The objective functions chosen reflect processing and transmission costs on the whole network. The constraints reflect quality of service requirements. The procedure is flexible and robust, since it allows multi-class traffic and poses no restrictions on the VP Network structure to be obtained.

One of the most formidable tasks in the search of the optimal VP Assignment is the calculation of blocking probabilities observed by the users. Since exact methods are practically not available, a set of approximation schemes is studied. The approximation schemes are all inspired by the reduced load approximation principles, and have varying computational complexities. The experience gathered from several comparisons with simulations

indicate that the Knapsack approximation performs accurately for our purposes. The results of the VPA Algorithm applied to a sample network show that the procedure runs successfully with different constraints and parameters. Increasing traffic loads under a constant blocking constraint forced the VPA Algorithm to use shorter VPs and increase sharing of links. Thus, the number of hops traversed by a connection has increased. Similarly, increasing the blocking constraint over same network conditions resulted in an increase on the number of hops traveled. The minimum hops and weighted minimum hops objectives have resulted in very close VP assignments. This may be due to the fairly balanced traffic in the network we have tested. The results of the experiments with maximum hop constraint (K) suggests the existence of local optima, and sensitivity of the result to the initial starting solution.

Areas for improvement and further study include incorporation of connection routing function into the optimization process, and the verification of the optimality of the resultant VP Network by investigating other search methods.

7 ACKNOWLEDGEMENTS

We would like to thank Dr. G. Bilbro for his helpful suggestions on the algorithm of section 4.

8 REFERENCES

- Addie, G. and Burgin, John and Sutherland, S. L. (1988) B-ISDN Protocol Architecture, in *Proceedings of IEEE GLOBECOM*, 22.6.1, Hollywood, FL.
- Arvidsson, A. (Oct. 1994) Real Time Management of Reconfigurable Virtual Path Networks in B-ISDN, *TRC Report No. 15/94*, Teletraffic Research Centre, The University of Adelaide.
- Aydemir, M. and Viniotis I. (1995) Virtual Path Assignment Problem in ATM Networks, *CACC Report*, NCSU, Raleigh, NC.
- Baskett, F. and Chandy, K. M. and Muntz, R. R. and Palacios, F. G. (1975) Open, Closed and Mixed Networks of Queues with Different Classes of Customers, in *Journal of the ACM*, **22**, 248-60.
- Bertsekas, Dimitri and Gallager, Robert. (1992) *Data Networks, 2nd Edition*. 524-8. Prentice Hall.
- Burgin, John and Dorman, Dennis. (Sep. 1991) Broadband ISDN Resource Management: The Role of Virtual Paths, in *IEEE Communications Magazine*, 44-48.
- Cheng, Kwang-Ting and Lin, Frank Yeung-Sung. (1994) On the Joint Virtual Path Assignment and Virtual Circuit Routing in ATM Networks, in *Proceedings of the IEEE GLOBECOM*, 777-82.
- Chlamtac, Imrich and Faragó, András and Zhang, Tao. (May 1993) How to Establish and Utilize Virtual Paths in ATM Networks, in *Conference Record of the IEEE International Conference on Communications (ICC)*, Geneva, Switzerland.
- Chung, Sun-Ping and Ross, Keith W. (1993) Reduced Load Approximations for Multirate Loss Networks, in *IEEE Transactions on Communications*, **41(8)**, 1222-31.

- Kanada, T. and Sato, K. and Tsuboi T. (1987) An ATM Based Transport Network Architecture, in *IEEE COMSOC Int. Workshop on Future Prospects of Burst/Packetized Multimedia Communications*, Osaka, JAPAN.
- Kaufman, Joseph S. (1981) Blocking in a Shared Resource Environment, in *IEEE Transactions on Communications*, COM-29(10), 1474-81.
- Kelly, F. P. (1986) Blocking Probabilities in Large Circuit-Switched Networks, in *Advances in Applied Probability*, 18, 473-505.
- Lam, S. S. (Jul. 1977) Queueing Networks with Population Size Constraints, in *IBM Journal of Research and Development*, 370-8.
- Lin, Frank Yeung-Sung and Cheng, Kwang-Ting. (Dec. 1993) Virtual Path Assignment and Virtual Circuit Routing in ATM Networks, in *Proceedings of the IEEE GLOBECOM*, 436-41. Houston, Texas.
- Sato, Ken-Ichi and Ohta, Satoru and Tokizawa, Ikuo. (1990) Broad-Band ATM Network Architecture Based on Virtual Paths, in *IEEE Transactions on Communications*, COM-38(8), 1212-22.
- Siebenhaar, Rainer (1994) Optimized ATM Virtual Path Bandwidth Management Under Fairness Constraints, in *Proceedings of the IEEE GLOBECOM*, 924-8.
- Tokizawa, I and Kanada, T. and Sato, K. (1988) A New Transport Network Architecture Based on Asynchronous Transfer Mode Techniques, in *Proceedings of ISSLS'88*, 11.2.1-5, Boston, MA.
- Whitt, Ward. (1985) Blocking When Service is Required From Several Facilities Simultaneously, in *AT&T Technical Journal*, 64(8), 1807-56. Prentice Hall Inc., NJ.

9 BIOGRAPHY

Metin Aydemir received the B.S.E.E. degree from Middle East Technical University, Ankara, Turkey in 1981. He continued to study towards a master's degree at Electrical Engineering Dept. in Florida Institute of Technology, Melbourne, Florida (1985), and worked in telecommunications software design area in ITT and SIEMENS. Since 1992, he is a PhD candidate in Electrical and Computer Eng. Department at NCSU, Raleigh, NC. His research interests are in high-speed network analysis and modeling, queueing networks, and optimization algorithms.

Yannis Viniotis received the B.Sc. degree from the University of Patras, Greece, and M.S. and Ph.D. degrees in Electrical Engineering from the University of Maryland, College Park. He is currently an Associate Professor of Electrical and Computer Engineering at North Carolina State University, Raleigh, NC. His research interests are in computer communication system design and analysis, with particular emphasis on Quality of Service, multimedia, multicasting and adaptive network control algorithms. He has published over 30 articles and lectured extensively on these topics. He was a guest editor of the *Performance Evaluation Journal*, in 1995, on High Speed Networks. He has chaired two international conferences on networking, in 1992 and 1993.